Tuesday 29th of Mar 2022

PYTHON WORKSHOP

Python Fundamentals

Prepared by the Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, UAE

Workshop Overview

The workshop aim

The workshop aims to teach the candidates the python language, starting from basics to advanced topics and consequently be able to apply the knowledge gained in application.

The workshop Duration

12 hours (6 Days)

Workshop Objectives

- Understand python basics.
- Learn to write and test Python 3 code.
- Dive into more advanced topics of python 3.
- Build voice recognition application in python.

Workshop Agenda

Day 1

- Workshop Overview
- Project Demo
- Introduction to Python
- Starting with VSC
- Variables and simple data types
- Comment and Print

- String manipulation
- If Statements
- Loops
- Lists
- Dictionaries
- Functions
- File Manipulation
- Final Project

Day 2

Day 3

Day 4

Day 5 & 6

Day 1 Outlines

- Workshop Overview
- Project Demo
- Introduction to Python
- Starting with VSC
- Variables and simple data types
- Comment and Print

1. Project Demo

Voice Assistant Program using Python Libraries

2. Python Basics

Fundamental Concepts and practices of python language using Visual Studio Code

Introduction to Python (1 of 2)

What Python ?

Python is one of the most popular **general-purpose cross-platform** programming languages.

Python applications

The domain of **Artificial intelligence** and **data science** is the most recent use of python. In addition, python was involved in many **other applications** development, such as : Graphic design, image processing, Games, Scientific/ computational Applications, Web frameworks, and Enterprise and Business applications.

Organizations using Python

- Google(Components of Google spider and Search Engine)
- Yahoo(Maps)
- YouTube
- Microsoft , and many others.

Introduction to Python (2 of 2)

Why to use Python ?

- Easy to write, read and understand.
- Interpreted Language
- Dynamically Typed
- Free and Open-Source
- Vast Libraries Support
- Assist developers in making **Machine learning models and solving computational problems**

Python Version

The latest released Python version was in **March 2022** which is **Python 3.10.4.**

3. Starting with VSC

Visual Studio Code IDE

Visual Studio Code

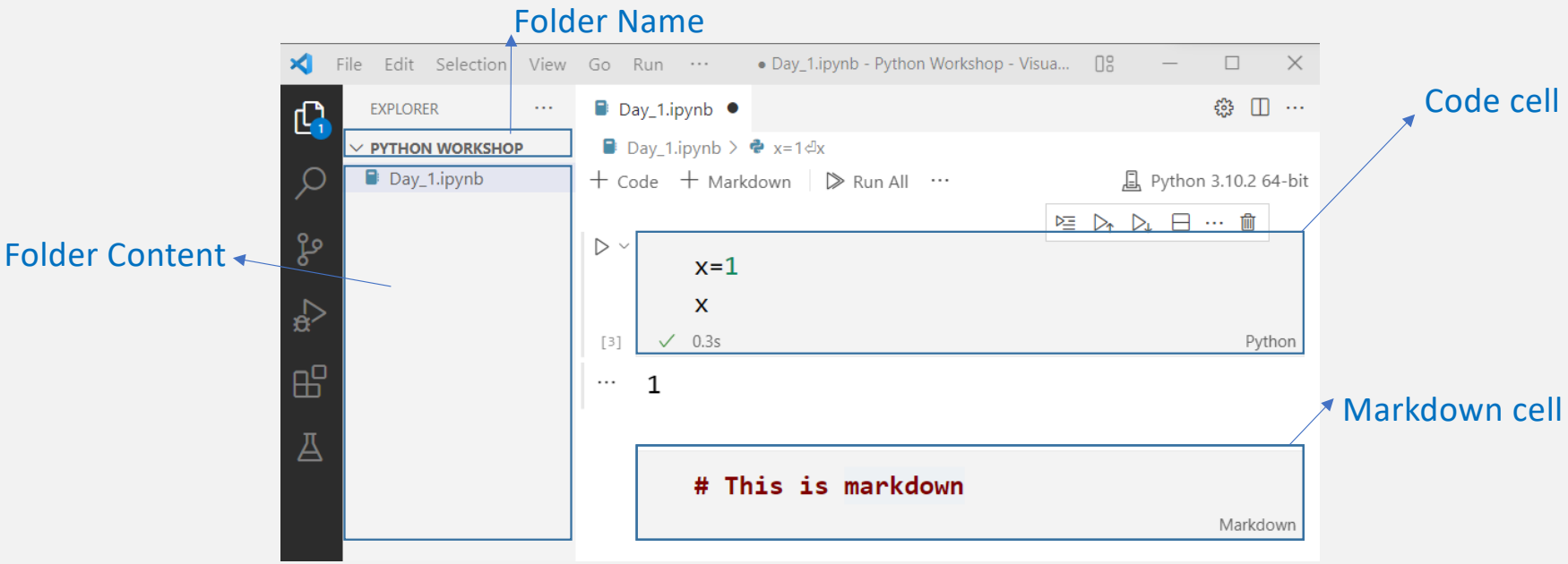
What is IPYNB

An IPYNB file is a notebook file used by Jupyter Notebook. Jupyter Notebook is an open-source web application that allows sharing documents with live Jupyter code, narrations, and visualization on the internet. The file is named IPYNB because Jupyter Notebooks was formerly named IPython Notebooks.

Why to use Visual Code Studio

- Simple and easy to use
- Free, open source, cross platform
- Useful Extensions

Visual Studio Code



python

4. Variables and simple data types

How to declare and fill-in different types of variables

Variables: Data Types & Casting

- **Numerical types**
  - `int`
  - `float`
- **Text type**
  - `str`
- **Boolean type**
  - `bool`
- **Sequence types**
  - `List`
- **Mapping type**
  - `Dictionary`

python

Numerical types: int

Declare int variable

Variables of int types are created when you assign a value to them

```
X = 1 # int
```

```
Print(type(X))
```

<class 'int'>

Convert to int type

You can convert variable type to int type using the int method

```
X = 2.3 #float
```

```
X = int(X)
```

```
Print(type(X))
```

<class 'int'>

Numerical types: float

Declare float variable

Variables of float types are created when you assign a value to them

```
y = 2.8 # float
```

```
Print(type(y))
```

<class 'float'>

Convert to float type

You can convert variable type to float type using the float method

```
y= 1 #int
```

```
y = float(y)
```

```
Print(type(y))
```

<class 'float'>

Text Types : Str

Declare str variable

Variables of str types are created using double (" ") or single quotes (' ')

```
S  = "hello world"
```

```
Print(type(y))
```

<class 'str'>

Convert to str type

You can convert variable type to str type using the str method

```
y= 1 #int
```

```
y = str(y)
```

```
Print(type(y))
```

<class 'str'>

python

Boolean Types : bool

Declare bool variable

Variables of bool types are created when you assign a value "False" or "True"

```
S  = True
```

```
Print(type(y))
```

<class 'bool'>

Convert to bool type

You can convert variable type to bool type using the bool method

```
y= "str variable "
```

```
y = bool(y)
```

```
Print(type(y))
```

<class 'bool'>

python

Sequence Types : List

Declare empty list variable

Sometimes you need to create an empty List which can be done using the constructor or empty parenthesis

```
#type 1: using empty parenthesis
list1  = []
```

```
#type 2 : using constructor
List1 = list()
```

Sequence Types : List

Create/Fill list variable

Sometimes you need to fill an empty List or directly create & fill a list which can be done using append, extend, or []

```
#Type 1: Filling empty list
list1  = []
list1.append("Apple")
list1.append(False)
list1.append(1.2)
# or list1.extend(["Apple", False, 1.2])
```

```
#type 2 : Create and filling List directly
list1 = ["Apple", False, 1.2]
```

python

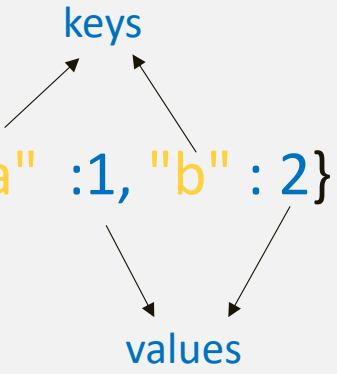
Mapping Types : Dict

A dictionary is a collection of key and value tuples which is ordered, changeable and do not allow duplicates.

Declare empty dict variable

```
dict1  = {}
```

```
dict1  = dict()
```



dict1

python

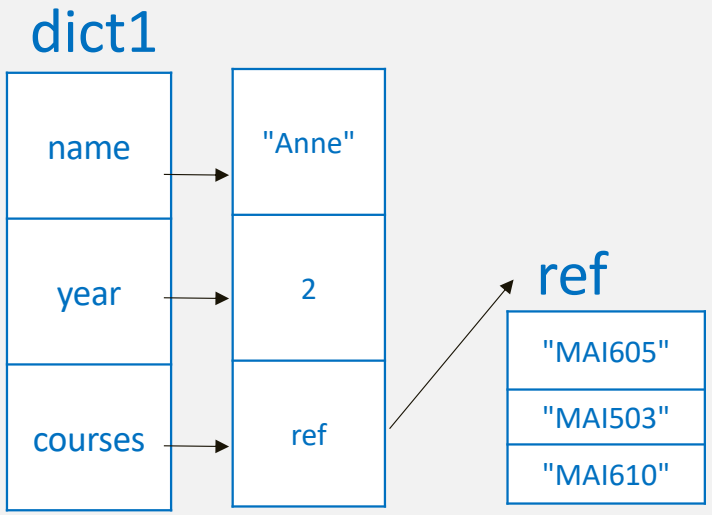
Mapping Types : Dict

Create/Fill empty dict variable

Sometimes you need to fill an empty List or directly create & fill a list which can be done using append, extend, or []

```
#type 1: Filling empty Dictionary
dict1  = []
dict1["name"] = "Anne"
dict1["year"] = 2
dict1["courses"] = ["MAI605","MAI503","MAI610"]
```

```
#type 2 : Create and filling List directly
dict1 = {"name":"Anne",
"year":2,"courses": ["MAI605","MAI503","MAI610"]}
```



Global and Local Variables

Local variable

Defined inside a function and its scope is limited to that function only

```
def f():

    s = "This is the Python Workshop"
    print("Inside Function:", s)

f()
print(s)
```

Global variable

are not defined inside any function and have a global scope

```
def f():
    global s
    s = "the string value has been updated"
    print(s)

s = "initial value of string"
print(s)
f()
```

5. Comment , Get user input and Print

Add declaration for your code, Interact with the user and use print function

Comment

- A Single line → Hash sign

```
# This is a comment
```

- Block of lines → Triple quotes

```
'''
This is a multiple line comment
This is a multiple line comment
This is a multiple line comment
'''
```

Interact with the user

- Basic Syntax

```
input(Message)
```

- Example

```
User_Name=input("Please Enter Your Name : ")
```

Print Hello World !

```
print("Hello World!")
```

Print Strings

Option 1

```
print("Hello my name is",Name)
```

Option 2

```
print("Hello my name is "+Name)
```

Option 3

```
print("Hello my name is {}".format(Name))
```

Option 4

```
print(f"Hello my name is {Name}")
```

Print Numbers

Option 1

```
print("My SID is",Number)
```

Option 2

```
print("My SID is "+str(Number))
```

Option 3

```
print("My SID is {}".format(Number))
```

Option 4

```
print(f"My SID is {Number}")
```

New line and separator

New line

```
print("The First line\n","The Second line")
```

Separator

```
print("The First line","The Second line", sep="--")
```

python